ARMY RESEARCH LABORATORY

An Implementation of Physical Layer Authentication Using Software Radios

by Paul Yu, John Baras, and Brian Sadler

ARL-TR-4888   **July 2009**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-4888 · **July 2009**

An Implementation of Physical Layer Authentication Using Software Radios

Paul Yu, John Baras, and Brian Sadler
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
July 2009	Summary	June to December 2008

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
An Implementation of Physical Layer Authentication Using Software Radios	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)	5d. PROJECT NUMBER
Paul Yu, John Baras, and Brian Sadler	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Research Laboratory ATTN: RDRL-CIN-T 2800 Powder Mill Road Adelphi MD 20783-1197	ARL-TR-4888

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Authentication is the process where claims of identity are verified and is a critical first step for sensitive communications. We propose a physical layer authentication technique that identifies radios based on their unique signal characteristics. The transmit signal is superimposed with a secret message-dependent authentication tag for the receiver to detect and validate. We present experimental results that indicate the usefulness of this technique: it has low impact on packet error, gives high quality authentication decisions, and is resistant to collisions.

15. SUBJECT TERMS

Software radio, authentication, security

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Paul Yu
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	30	19b. TELEPHONE NUMBER *(Include area code)* (301) 394-1722
Unclassified	Unclassified	Unclassified			

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Authentication is the first step in secure communications. The failure to properly authenticate users can result in serious damage since the adversary can do what any valid user can do. There have been many accounts of malicious insiders leaking or destroying sensitive information.

An authentication system verifies the identity of valid transceivers with high probability while it accepts invalid transceivers with very low probability. Message authentication is typically performed by time multiplexing authentication tags with the data stream (figure 1) (*1*). The receiver verifies that the tag corresponds to the data in order to authenticate the transmitter. However, this approach always reduces the data bandwidth because data transmission is periodically halted to transmit authentication information.

We shift the paradigm towards superposition: how can authentication be performed by transmitting data and authentication *simultaneously* (figure 1b)? In the following, we describe the authentication framework and how the tags are formed. We then detail the experiments we performed to validate the theory using a software radio platformed called GNU radio. The results of the experiment demonstrate the effectiveness of the technique: it is stealthy to adversarial detection, robust to noise, and secure from common authentication attacks.
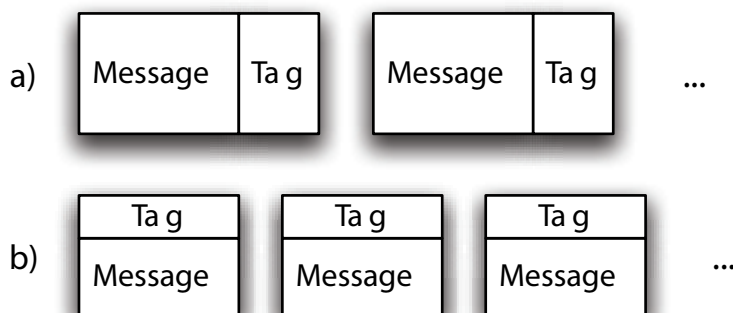
Figure 1. Time multiplexed (a) vs. superimposed (b) authentication.

2. Physical Layer Authentication

2.1 Requirements and Definitions

We introduce terminology that is useful when describing physical layer authentication systems. Signals that contain superimposed authentication tag are called *tagged*; those that do not are called *untagged*. Transceivers that are aware of the physical layer authentication system are called *aware*; those oblivious to it are called *unaware*.

In any security system, there are three main parties in play: the transmitter, receiver, and adversary. It is helpful to introduce assigned names to these parties, call them Alice, Bob, and Eve, respectively. Therefore, we have the situation where Alice sends authenticated messages to Bob while Eve tries to frustrate their efforts (figure 2).
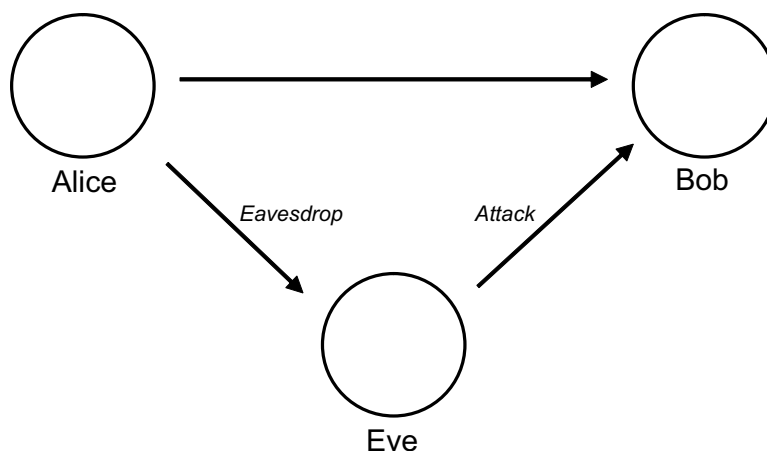
Figure 2. Alice (transmitter), Bob (receiver), and Eve (adversary).

We want the physical layer authentication system to possess the following properties:

1. Stealth: A system has stealth when the authentication tag has little impact on the operation of the underlying system. Practically speaking, it does not effect the packet error rate. Another aspect of stealth is in its ability to be hidden from adversaries who wish to simply decide on whether authentication is present.

2. Robustness: A system has robustness when the intended received is able to reliably authenticate the transmitter.

3. Security: A system has security when it is difficult for an adversary to successfully attack the system, either by impersonating the transmitter or by substituting messages.

While security is a common requirement for authentication systems, stealth and robustness are requirements unique to physical layer systems. In time multiplexed authentication systems, since the authentication is transmitted in plain view, there is little that can be done to hide or improve the robustness of the authentication. We will discuss the implications of the properties as well as the metrics we use to compare them.

2.2   Motivation and Rationale

Our approach to authentication at the physical layer perturbs the transmission signals in a controlled manner to uniquely identify the transmitter. The perturbations over a block of data symbols (call it a packet) form an authentication tag. The receiver searches for the authentication tag during observation. If it is found, the receiver decides that the transmitter is authentic. If it is not, the receiver decides that the transmitter is not authentic.

The benefits of authentication framework are many:

1. At no time is data transmission halted, as can be seen in figure 1. This potentially increases data throughput. However, in order to fix the total transmission power, some power is allocated away from data to authentication, thereby decreasng the data signal-to-noise ratio (SNR). Thus, although the data is constantly transmitted in this scheme, the reliability of the symbols may worsen due to decreased signal strength. Therefore, the system should be stealthy to limit the degradation of the data SNR.

2. Since the method resides entirely at the physical layer, the higher layers do not need to be modified in order to add authentication. This is of particular interest for existing systems where no provision for authentication currently exists. Rather than defining a new packet structure that all nodes must follow, the authentication tags are concurrently transmitted at low power so that while the legacy receivers continue to operate as normal, new receivers have the additional capability to authenticate the transmitter.

3. It is possible to make the authentication difficult to detect. Fundamentally, it is easier to attack a target when it is in plain view than when it is hidden. With the time multiplexed scheme (figure 1a), the adversary knows that the authentication is always present and thus can mount strong attacks in order to disrupt the authentication *(1)*. However, if the tags were superimposed and transmitted at low power, the adversary cannot say with certainty whether or not the authentication is present and therefore the attacks are less likely to succeed.

4. Even if the adversary knows that the authentication is present, the presence of noise works to obscure it from tampering. Typical attacks on authentication tags involve

either exploiting weaknesses in the tag generation function (e.g., hash or other one-way function) or by brute force ($2-4$). While these attacks are still possible, success is potentially more difficult to achieve because the observations are fundamentally noisy and it is not easier to decrypt a noisy message than a noiseless one. Thus, the physical layer approach can increase the challenge to the adversary without exposing any new vulnerabilities to the underlying cryptographic primitives.

We now turn our attention to how the authentication tags are formed and superimposed with the messages.

2.3 Framework for Physical Layer Authentication

Suppose that Alice and Bob communicate using narrowband signals modulated over a single carrier. We assume that the message and tags are independent and identically distributed, and therefore, we do not use time indices. Denote a message by $\mathbf{b}$. Alice codes and modulates the message to form the message signal $\mathbf{s} = f_e(\mathbf{b})$. The authentication signal $\mathbf{t}$ is formed by

$$\mathbf{t} = g(\mathbf{b}, k), \tag{1}$$

where $g(\cdot)$ is a function known to Alice, Bob, and Eve, and k is a secret key known only to Alice and Bob. We require that $g(\cdot)$ be collision resistant so that when $\mathbf{b} \neq \mathbf{b}'$ and $k \neq k'$, $g(\mathbf{b}, k) = g(\mathbf{b}', k')$ with negligible probability. This requirement states that with high probability, a tag generated from different keys and messages will be different. If this were not the case, the authentication tags would have little value because they could be associated with multiple messages or keys.

Now we assume that $\mathbf{s}$ and $\mathbf{t}$ are uncorrelated and have equal power. Though message and tag are not independent, they can be made nearly uncorrelated through careful selection of $g(\cdot)$. The transmitted signal is a scaled superposition of message and tag:

$$\mathbf{x} = \rho_s \mathbf{s} + \rho_t \mathbf{t}, \tag{2}$$

where $\rho_s^2 + \rho_t^2 = 1$. Note that no authentication is transmitted when $\rho_s = 1$ and that $\mathbf{x}$ always has the same power regardless of the choice of ρ_s, ρ_t.

Bob observes the signal through a noisy channel

$$\mathbf{y} = h\mathbf{x} + \mathbf{w}, \tag{3}$$

where h is a complex scalar representing the channel attenuation and $\mathbf{w}$ is additive white Gaussian noise (AWGN). With his channel estimate $\hat{h}$, he estimates the message signal and

uses $f_d(\cdot)$ to recover the message:

$$\hat{\mathbf{x}} = \frac{\hat{h}^*}{|\hat{h}|^2}\mathbf{y} \tag{4}$$

$$\hat{\mathbf{b}} = f_d(\hat{\mathbf{x}}) \tag{5}$$

where $f_d(\cdot)$ satisfies $z = f_d(f_e(z))$ for all z.

With the estimated message Bob can recreate the tag because he has the secret key k:

$$\hat{\mathbf{t}} = g(\hat{\mathbf{b}}, k) \tag{6}$$

He then searches for the tag in the observed signal. Bob first removes the message signal and then match filters the residual $\mathbf{r}$ with the estimated tag $\hat{\mathbf{t}}$ to form his test statistic τ:

$$\mathbf{r} = \frac{1}{\rho_t}\left(\hat{\mathbf{x}} - \rho_s f_e(\hat{\mathbf{b}})\right) \tag{7}$$

$$\tau = \hat{\mathbf{t}}^*\mathbf{r} \tag{8}$$

The statistic τ is then used to determine authenticity by performing a threshold test with a certain false alarm probability α. The hypotheses are as follows:

$$H_0 \;\; : \;\; \mathbf{r} \text{ does not contain } \hat{\mathbf{t}} \text{ (the message is not authentic)} \tag{9}$$

$$H_1 \;\; : \;\; \mathbf{r} \text{ contains } \hat{\mathbf{t}} \text{ (the message is authentic)} \tag{10}$$

Bob makes the decision H_δ, where

$$\delta = \begin{cases} 0 & \tau < \tau^0 \\ 1 & \tau \geq \tau^0 \end{cases} \tag{11}$$

where τ^0 is set to satisfy the false alarm probability α:

$$p(\delta = 1|H_0) = \alpha \tag{12}$$

3. Radio Design

The authentication scheme was implemented on GNU radio, a software-defined radio (SDR) platform that is in active development by the open source community. The primary concept of a software radio is to have the software as close as possible to the antenna, as shown in figure 3. Compared with traditional radios where modulations and codes are defined with special circuitry, SDR shifts the computational load from hardware to
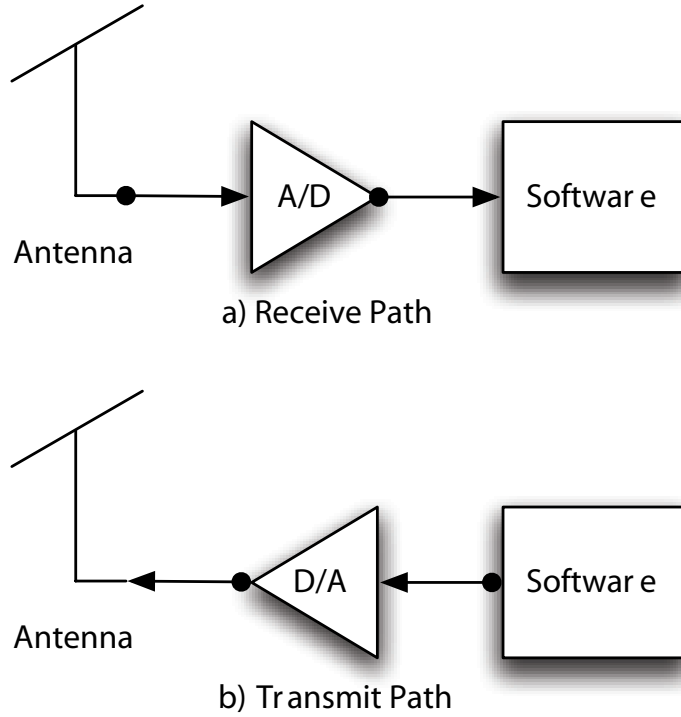
Figure 3. A fundamental concept of SDR is the placement of software as close as possible to the antennae. Only an analog-to-digital converter (ADC) separates the software from the antenna in the receive path (a), while a digital-to-analog (DAC) is present in the transmit path (b).

software. With the increase in processing power and the associated decrease in cost, SDR is becoming a more and more viable solution for powerful and adaptable radios. Practically speaking, the SDR paradigm increases the speed and ease of prototyping, testing, and configuring new radios.

For our experiment, we do not modify the hardware, but we make extensive software-side extensions to implement the authentication. However, since the hardware imposes limitations on the authentication *(5)*, we first detail the relevant hardware specifications. Then, we detail our software implementation of the physical layer authentication scheme over the GNU radio platform.

3.1 Hardware Capabilities

The software interfaces with the radio transceiver via a universal serial bus (USB) interface. The radio transceiver in our experiment is the Universal Software Radio Peripheral (USRP, pronounced *"usurp"*), which is the most popular and commonly available peripheral used by the GNU radio project. As seen in figure 4, the USRP consists of a USB interface, a

field-programmable gate array (FPGA), ADCs and DACs, and daughterboards. The daughterboards are responsible for the frequency tuning and conversion between intermediate (IF) and radio frequencies (RF), and are swappable for flexible configuration. In the following, we detail the signal receive path to highlight the design of the hardware.
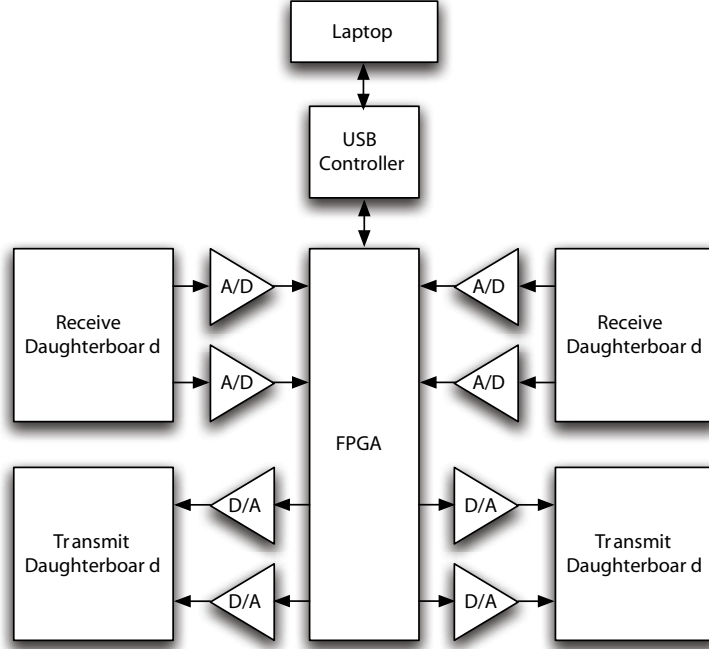

Figure 4. An overview of the hardware setup: the laptop is connected via USB to the USRP. The USRP consists of an FPGA responsible for up/down conversions, ADCs and DACs, and various plug-in daughterboards.

3.1.1   Daughterboard RFX2400

The signal is captured by an antenna attached to an RFX2400 daughterboard. The RFX2400 is a 2.3–2.9 GHz band transceiver with a 20 MHz transmit/receive bandwidth. The received signal passes through a mixer to downcovert the signal to the IF*. Then, the signal is amplified up to 70 dB via automatic gain control (AGC) before being sent to the USRP motherboard.

---

*By converting signals to an IF rather than going directly between RF and baseband, the quality of the circuit can be vastly improved (by allowing use of crystal filters, for example). Receivers which do this are called superheterodyne for their use of the heterodyne principle, which is based on the identity $2\sin(\theta)\sin(\phi) = \cos(\theta - \phi) - \cos(\theta + \phi)$.

3.1.2 USRP

The USRP board has four 12-bit ADCs that are capable of processing up to 64 mega-samples per second from the daughterboards. Depending on configuration, these channels may contain either real or paired in-phase (I) and quadrature (Q) samples.

The digitized samples are then sent to the FPGA[†]. The FPGA uses a numerically controlled oscillator (NCO) to convert the samples from IF to baseband. Then cascaded integrator-comb (CIC) filters are used to decimate the oversampled signal to lower the data rate. This paring down is the digital down conversion (DDC) and is necessary for transmission over the USB 2.0 interface. The resultant total bandwidth over all channels is limited to 32 megabytes/s: 16-bit signed integers in I/Q format, i.e., 4-bytes per complex sample at 8 megasamples/s. Of course, lower bandwidths are possible by setting the decimation factor, e.g., 64 MHz/250 = 256 kHz. Finally, the samples are transmitted to the computer via USB.

The transmit path is essentially the reverse of the receive path. Digital samples arrive at the USRP via USB and are interpolated and up-converted to the IF. Then they are passed through DACs and sent to the daughterboard, where they are mixed to the RF, amplified, and transmitted over the antenna.

3.1.3 Laptops

We use two identical 2.0 GHz Pentium M laptops with 512 MB RAM. Each runs Ubuntu Linux 7.04 (Feisty Fawn) with the GNU radio software installed. The software is extended for physical layer authentication capability as described in section 3.2. One laptop controls the transmitter; the other controls the receiver.

3.2 Software Design

We modified the GNU radio platform to add authentication at the physical layer. It was written with a combination of C++ and Python for a good tradeoff between processing speed and rapid prototyping. The signal processing blocks (e.g., filters, phase locked loops) were written in C++ and joined together in Python.

For this experiment, we modified existing signaling blocks and also created our own. In the following, we detail the changes made to the transmit and receive paths.

[†]Altera Cyclone EP1C12 chip.

3.2.1 Transmitter

Figure 5 shows a diagram of the transmitter. The original system takes the payload and constructs a packet around it. It adds a preamble, access code, header, and cyclic redundancy check (CRC) (figure 6). The packet is then differential binary phase shift keying (DBPSK) modulated, pulse shaped, and transmitted.
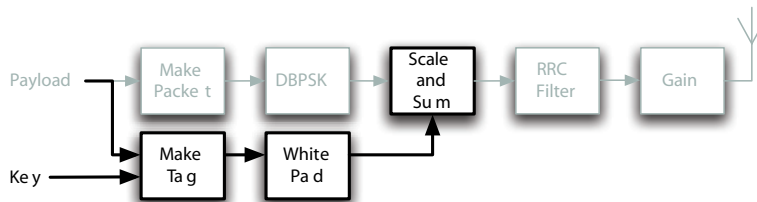

Figure 5. Transmitter signal path. Unmodified processing blocks are grayed out; modifications are darkened.

To implement the authentication, we made the following changes. We added a tag creation block that generates the authentication tag from the payload and a secret key. Then, the authentication tag was padded to align the tag with the message payload (figure 6). The message packet and padded tag are scaled and superimposed—the padding ensures that only the message payload is perturbed and that the important header information is untouched. In general, we may choose to perturb the entire packet. However, since the header may be used for synchronization or other important purposes, we chose not to alter it.
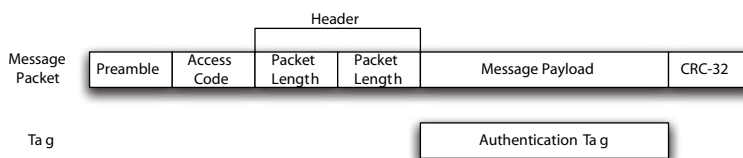

Figure 6. Packet format. Note that the tag has non-null information coincident with the packet payload; no other portion of the packet is modified by the superposition.

In our implementation, we use binary signaling for the authentication tag: we either increase or decrease the voltage of a payload symbol depending on each particular tag bit. This has the nice property of being easy to decode over DBPSK-modulated messages since the receiver only has to observe the symbol amplitude and not the symbol phase.

3.2.2 Receiver

Figure 7 shows a diagram of the receiver. The receiver performs AGC, root-raised cosine filtering, timing (Mueller and Muller algorithm), and phase (Costas loop) synchronization before DBPSK demodulation.
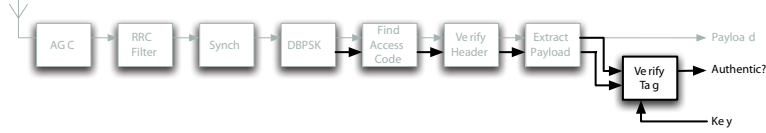
9

Figure 7. Receiver signal path. Unmodified processing blocks are grayed out; modifications are darkened.

In the original system, after digitization the receiver scans for the access code that indicates the beginning of a packet. After finding the access code, it then verifies the integrity of the header and extracts the payload with CRC. The payload is then checked with the CRC. If the redundancy check fails, the packet is discarded; otherwise, it is accepted.

To obtain the tag, we modified the DBPSK module to return not only the demodulated symbols but the sampled digital signal (i.e., the output of the ADC). The sampled signals are paired and together pass through each subsequent block until the payload is verified. If the payload passes the CRC check, the signals proceed to the tag detection block; otherwise, the samples are discarded and no further processing is done.

With a successful CRC check, the sampled signal arrives at the tag detection block. The verified payload (symbols) and the receiver's secret key are used to generate the authentication tag. The receiver takes the sampled signal corresponding to the payload (figure 6) and correlates it with the tag. When the correlation exceeds the threshold chosen to limit false alarms, the packet is deemed authentic and accepted.

4.   Testing Procedure and Results

The transmit and receive stations were placed approximately 20 ft apart without a line of sight. The transceivers operated at 2.44 GHz to avoid strong interference from the campus wireless network and cordless telephones.

The receiver continuously scans the channel for packets. The transmitter sends 48 k (4×2^{10}) packets at 500 kbps. We used two payload lengths: 128 and 192 bytes, of which 4 bytes are set aside as pilot symbols. For each packet length, we consider the following test scenarios (TS):

TS 1: The transmitter does not transmit any authentication.

TS 2: The transmitter superimposes the authentication on the packets but its secret key does not match that of the receiver.

10

TS 3: The transmitter superimposes the authentication on the packets and its secret key matches that of the receiver.

The receiver should reject the packets in cases 1 and 2, and only accept the packets in case 3. Accepting a packet in case 1 is the most innocuous false alarm. In case 2, accepting a packet leads to a security breach since the keys do not match. For cases 2 and 3, where the authentication is present, the experiments were repeated at $0.1, 0.2, 0.3, \ldots, 1.0\%$ authentication powers.

The following data sets (DS) were collected:

DS 1: Digitized signal samples

DS 2: Number of received packets (error-free)

DS 3: Number of authenticated packets

The interpretation of the data depends on the TS and is discussed in sections 4.1, 4.2, and 4.3. Figure 8 gives a preview of how the data was processed.

Figure 8. The TS and DS are used to evaluate the authentication system. The data collected in each TS is used to compute the stealth, robustness, or stealth metrics.

4.1   Stealth

We quantify the stealth of the authentication system based on the impact and presence of the authentication tags. These are measured by packet error rate and noise distribution, respectively. The packet error rate indicates the impact of the authentication on message recovery. The noise distribution indicates the detectability of the perturbation to the unaware receiver.

11

4.1.1 Impact

The impact of the authentication upon the receiver is found by comparing the number of packets received without error (DS 2) between the scenarios when the authentication is absent versus when it is present (TS 1 vs. TS 2 and 3). Since there are many factors THAT affect how packets are dropped (e.g., not detected, failed header check, failed CRC), which may be due to time variation of the channel, we repeat the experiment multiple times. The impact of the authentication is found by determining how adding the authentication causes the packet error rate increase.

The observed packet error rates are shown in figure 9. We observe no conclusive link between authentication power and packet error rate for the range of authentication powers tested. At such low authentication power, we suggest that the perturbation has a minimal impact and that the time-varying nature of the channel plays a much greater role on the packet error. This confirms the analysis in section 2.

Figure 10 is a snapshot of observed SNR across consecutive frames for various values of authentication power. The SNR values are obtained through the use of pilot symbols. We note that the SNR is not noticeably degraded with the addition of low power authentication tags; rather, the time variation of the channel plays a much larger role in the measured SNR. We calculate the 95% confidence interval that the received signal does not contain authentication. In this particular snapshot, the majority of the packet SNR in three cases fall inside the 95% confidence interval. For those SNRs that fall outside of the interval, most are actually false alarms (when no authentication is transmitted).

4.1.2 Presence

The previous section established that the packet reception is minimally impacted when the authentication is injected at low power. Now we turn our attention to the distortion that is observed by the receiver. For each packet that is received correctly, we record the amplitude distortion (DS 1).

We study the noise by calculating the empirical cumulative distribution function (CDF) of the amplitude distortion over thousands of packets. The baseline distribution yields the noise characteristics of the channel under normal conditions, i.e., when no authentication is transmitted.

The presence of the authentication system is hidden when the resultant noise distribution at the receiver is close to the baseline. When the noise distribution is not close, its presence can be discovered through the use of goodness of fit tests. Assume that the receiver knows the baseline noise CDF, perhaps through training with a known transmitter. The receiver
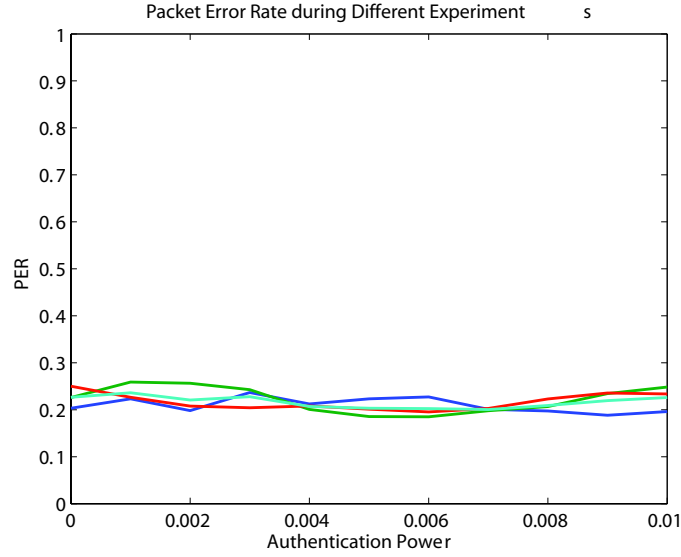
Figure 9. The packet error rate for various sample runs versus the power of the authentication signal. At low authentication powers, no significant deviation from the baseline packet error rate was observed. Each line represents a different test run.
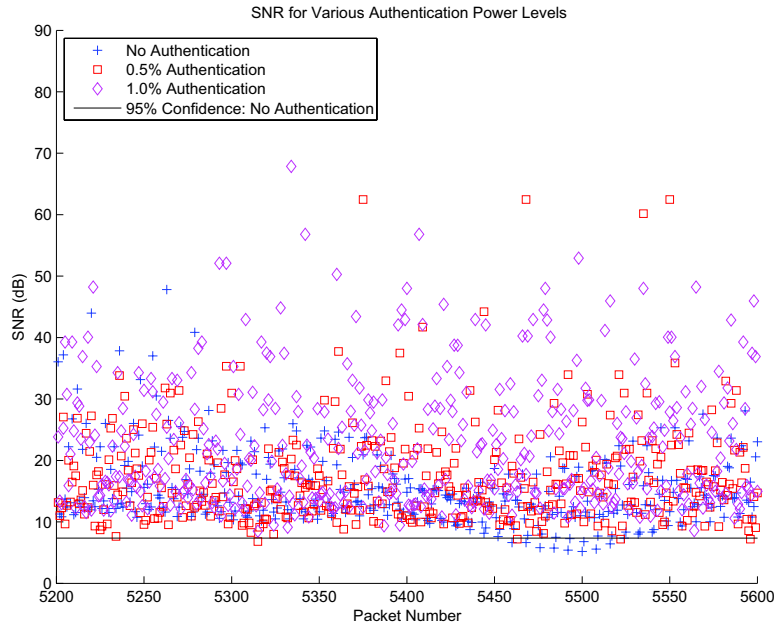

Figure 10. The observed SNR of tagged and untagged signals for a few consecutive packets. The majority of the packet SNR in three cases fall inside the 95% confidence interval for no authentication present in the signal; in this snapshot, most of those that fall outside of it are actually false alarms.

can compare the observed noise statistics with the baseline CDF in order to determine whether the signal is being perturbed. The tests operate at a user-specified probability of false alarm, which is usually set very low to return useful detections.

Figure 11 shows the CDF for some representative authentication powers. The CDF are further apart when the authentication power increases. In theory, goodness of fit tests will therefore reject the distributions as being unequal when the authentication power is too high.
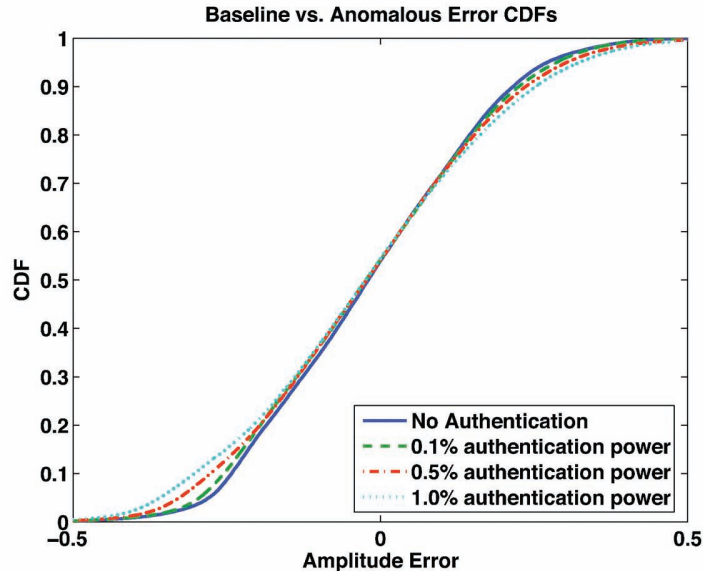

Figure 11. The observed CDF of the estimated noise for various authentication powers over thousands of packets. Larger authentication powers deviate more from the baseline CDF.

However, we found that the time-variation of the channels inhibits good performance of the tests. Using the Kolomogorov-Smirnov test with a 1% false alarm probability over a window of a few hundred packets, the receiver correctly flags the tagged signals as anomalous. However, the receiver also flags the untagged signals as anomalous—even though no authentication was being transmitted. For an idea of what the estimated CDF looks like over a few packets, see figure 12. Thus, the receiver needs more powerful techniques to discriminate between pure noise and perturbation in noise—the Kologorov-Smirnov (KS) test is not able to distinguish between tagged and untagged signals reliably.

The fact that it is difficult to discriminate using goodness of fit tests indicates that stealth may be further improved by time-multiplexing the authentication. That is, the receiver is faced with a more difficult anomaly detection problem when the authentication tags are injected into some, but not all, of the packets.

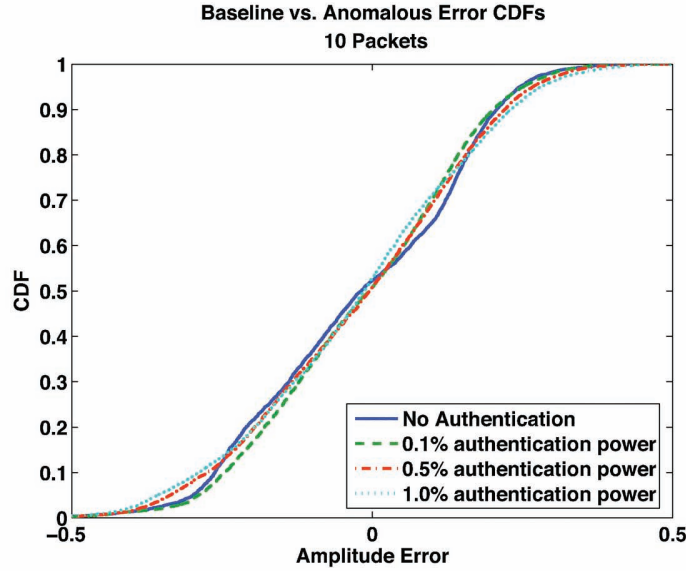**Baseline vs. Anomalous Error CDFs**
10 Packets

Figure 12. The observed CDF of the estimated noise for various authentication powers over 10 packets. CDFs of all depicted authentication powers differ; the CDF with no transmitted authentication does not match the long-term estimate shown in figure 11.

4.2 Robustness

We quantify the robustness of the authentication system by its authentication probability for a fixed false alarm probability. We have the transmitter and receiver share the same key (TS 3) and analyze the number of authenticated packets (DS 3) for both 128- and 192-byte payloads.

The detection probabilities are found in table 1. With the same power allocation, longer authentication tags have more energy and thus result in higher quality decisions. For example, increasing the payload from 128 to 192 bytes increases the detection probability from 39% to 97%.

Table 1. Authentication Probability.

	Tag power			
	0	0.001	0.005	0.010
L = 128	0.001	0.391	0.999	1.000
L = 192	0.001	0.973	1.000	1.000

The test statistics with 128-byte payloads are shown in figure 13(top) for various authentication powers. The experiment is repeated for 192-byte payloads as shown in
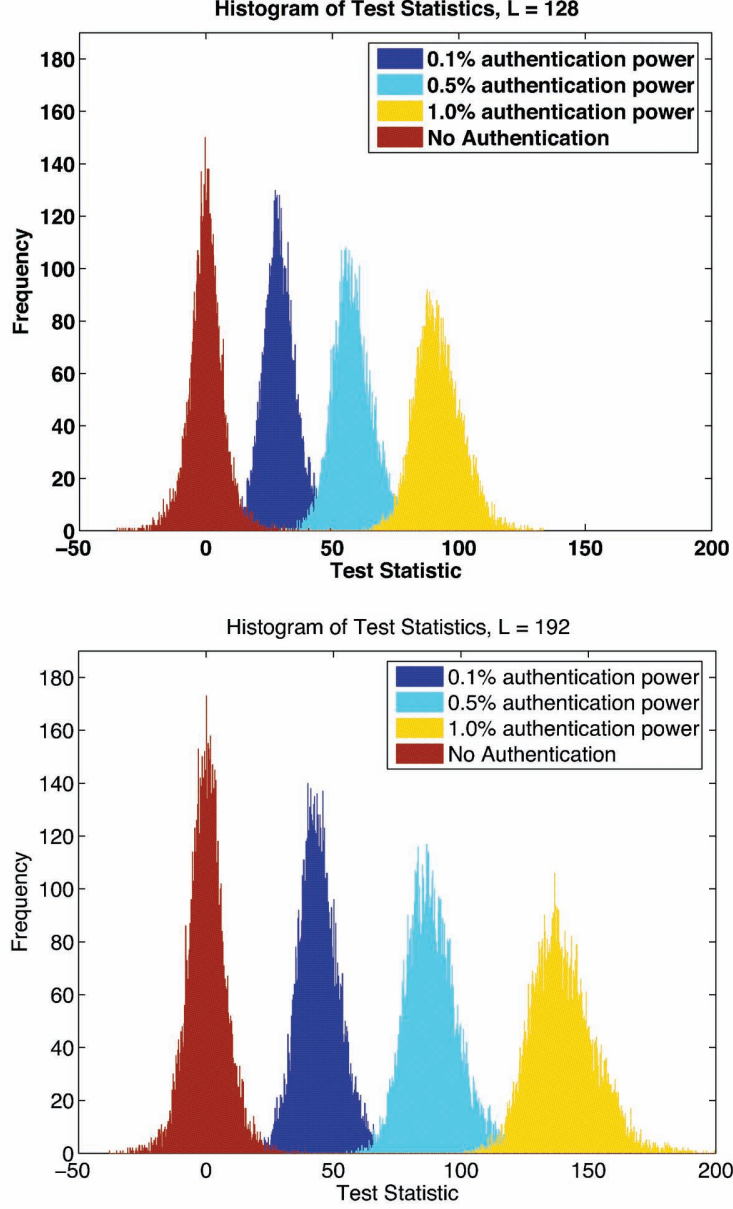
15

Figure 13. The observed CDF of the estimated noise for various authentication powers over 10 packets. CDFs of all depicted authentication powers differ; the CDF with no transmitted authentication does not match the long-term estimate shown in figure 11.

figure 13(bottom). The statistics are clearly separated from the untagged signal case (no authentication transmitted), even for very low authentication power. Increasing the perturbation length increases the energy, and hence the performance of the authentication improves as well. We see that there is a clear relationship between the energy of the authentication and its performance.

4.3 Security

We measure the security of the authentication system by observing the probability of falsely authenticating an invalid transmitter. That is, we compare the authentication probabilities (DS 3) between the scenarios where the receiver does not know the key (TS 2) versus when the receiver does know the key (TS 3).

For test scenario 2, the transmitter and receiver are seeded with different keys. The transmitter then sends authenticated packets. The receiver is able to decode the payload because of stealth (section 4.1), but should not accept the packet as authentic because the authentication is not generated using the same key.

Similarly, for TS 3, the transmitted and receiver are seeded with identical keys, and the receiver should authenticate the packets.

In our tests, we did not observe any false positives in TS 2, while the authentication performed as usual in TS 3 (as in the robustness tests). We were unable to perform an exhaustive test covering all possible keys for all possible payloads so we cannot conclusively state how secure the authentication system is through this test. However, it does lend some evidence to the theoretical security analysis *(6)*.

5.   Conclusions

We have described experimental results obtained via software radios operating at the 2.44 GHz center frequency. With out experiment, we are able to demonstrate that our scheme is physically realizable and offers good results with low complexity. We observe that the time variation of the channel inhibits the ability of the adversary to distinguish between tagged and untagged signals, especially when the authentication power is low. that is, the hypothesis has low power when the false alarm probability is reasonably low. The results of hte experiments detailed above indicate that outside of the simulation environments, implementations of this authenticatioin scheme may experience better than expected stealth.

References

[1] Menezes, A. J.; van Oorschot, P. C.; Vanstone, A. *Handbook of Applied Cryptography.* CRC Press, 2001.

[2] Chabaud, F.; Joux A. Differential Collisions in SHA-0. *Proceedings of the CRYPTO '98*, Aug 1998, 56–71.

[3] Simmons, G. J. A Survey of Information Authentication. *Proc. IEEE* **May 1998**, *76* (5), 603–620.

[4] Bellare, M.; Kohno, T. Hash Function Balance and Its Impact on Birthday Attacks. *Advances in Cryptology–EUROCRYPT '04*, 2004, 401–418.

[5] Yu, P. Physical Layer Authentication. Ph.D. dissertation, University of Maryland, College Park, 2008.

[6] Yu, Y.; Baras, J. S.; Sadler, B. M. Physical Layer Authentication. *IEEE Trans. Inf. Forensics Security* **March 2008**, *3* (1), 38–51.

List of Symbols, Abbreviations, and Acronyms

ADC	analog-to-digital converter
AGC	automatic gain control
AWGN	additive white Gaussian noise
CDF	cumulative distribution function
CIC	cascaded integrator-comb (filter)
CRC	cyclic redundancy check
DAC	digital-to-analog donverter
DBPSK	differential binary phase shift keying
DDC	digital down conversion
DS	data sets
FPGA	field-programmable gate array
I	in-phase
IF	intermediate frequency
KS	Kolomogorov-Smirnov (test)
NCO	numerically-controlled oscillator
Q	quadrature
RF	radio frequency
SDR	software defined radio
SNR	signal-to-noise ratio
TS	test scenarios
USB	universal serial bus
USRP	Universal Software Radio Peripheral